



HOWTO Install Raptor MUA+WEB on Debian Linux

14 Aprile 2015

Autore: Marco Malavolti

Indice

1) Introduzione.....	3
2) Pacchetti richiesti.....	3
3) Configurare SSL per Apache2.....	4
4) Installazione.....	6
4.1) Scaricare e Installare il Raptor MUA e il Raptor WEB.....	6
4.2) Configurare il Raptor MUA e WEB.....	7
4.3) Scambio di Chiavi tra MUA e ICA.....	12
4.3.1) Raptor ICA - Importazione Chiave del MUA.....	12
4.3.2) Raptor MUA - Importazione Chiave del ICA.....	12
5) Appendice A.....	13

1 Introduzione

Questo è un tutorial per gli utenti che vogliono installare un Raptor Server (MUA+WEB) su una macchina Debian Linux (Wheezy).

Il Raptor Server ha il compito di raccogliere e presentare le informazioni fornite dai vari Raptor Client (ICA) ad esso collegati.

2 Pacchetti richiesti

- ntp
- vim
- java (openjdk o sun-java)
- alien
- wget
- mysql-server

3 Configurare SSL per Apache2

1. Diventare ROOT sulla macchina prescelta:
 - `sudo su -`
2. Creare la cartella “/root/certificates” e inserire in essa il certificato, la chiave e la Terena-chain consegnatevi da GARR:
 - `mkdir /root/certificates`
 - `mv /tmp/cert-server.pem /tmp/key-server.pem /tmp/Terena-Chain.pem /root/certificates`
3. Modificare i file di configurazione di Apache2:
 - `vim /etc/apache2/sites-available/default-ssl`

```
<VirtualHost _default_:443>
    ServerName raptor.example.org:443
    ServerAdmin admin@example.org
    ErrorLog ${APACHE_LOG_DIR}/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    SSLEngine On
    SSLProtocol all -SSLv2 -SSLv3 -TLSv1
    SSLCipherSuite "KEDH+AESGCM:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES256-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA256:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA256:AES256-GCM-SHA384:!3DES:!DES:!DHE-RSA-AES128-GCM-SHA256:!DHE-RSA-AES256-SHA:!EDE3:!EDH-DSS-CBC-SHA:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC-SHA:!EDH-RSA-DES-CBC3-SHA:!EXP-EDH-DSS-DES-CBC-SHA:!EXP-EDH-RSA-DES-CBC-SHA:!EXPORT:!MD5:!PSK:!RC4-SHA:!aNULL:!eNULL"
    SSLHonorCipherOrder on

    # Disable SSL Compression
    SSLCompression off

    # Enable HTTP Strict Transport Security with a 2 year duration
    Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"

    SSLCertificateFile /root/certificates/cert-server.pem
    SSLCertificateKeyFile /root/certificates/key-server.pem
    SSLCertificateChainFile /root/certificates/Terena-Chain.pem

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>

    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>

    BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

    # MSIE 7 and newer should be able to use keepalive
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
```

```
</VirtualHost>
```

- `vim /etc/apache2/ports.conf`

```
#NameVirtualHost *:80  
#Listen 80
```

4. Abilitare SSL su Apache e riavviare il servizio:
 - `cd /etc/apache2/sites-available`
 - `a2enmod header ; a2enmod ssl ; a2ensite default-ssl`
 - `service apache2 restart`
5. Verificare di ottenere almeno un grado “A” sul sito:
 - <https://www.ssllabs.com/ssltest/>

4 Installazione

4.1 Scaricare e Installare il Raptor MUA e il Raptor WEB

1. Diventare ROOT sulla macchina prescelta:
 - `sudo su -`
2. Installare i pacchetti richiesti:
 - `apt-get install ntp vim openjdk-7-jre alien wget`
3. Scaricare il pacchetto del Raptor ICA in `/usr/local/src`:
 - `wget -c -nd`
`http://download.opensuse.org/repositories/home:/rhyssmith:/raptor/SLE_11/noarch/raptor-web-1.2.1-1.1.noarch.rpm -O /usr/local/src`
 - `wget -c -nd`
`http://download.opensuse.org/repositories/home:/rhyssmith:/raptor/SLE_11/noarch/raptor-mua-1.2.1-1.1.noarch.rpm -O /usr/local/src`
4. Convertire i pacchetti RPM in pacchetti DEB con **alien**:
 - `cd /usr/local/src`
 - `alien --scripts raptor-web-1.2.1-1.1.noarch.rpm`
 - `alien --scripts raptor-mua-1.2.1-1.1.noarch.rpm`
5. Installare il pacchetto DEB creato:
 - `dpkg -i raptor-web_1.2.1-2.1_all.deb`
 - `dpkg -i raptor-mua_1.2.1-2.1_all.deb`

4.2 Configurare il Raptor MUA e WEB

1. Modificare il file `/etc/init.d/raptormuad` aggiungendo i valori in grossetto:

```
1 #!/bin/bash
...

25 ### END INIT INFO
26
27 ##
28 ## Added the JAVA_HOME detection, the Debian way --csr2 2012-07-19
29 ##
30 # The first existing directory is used for JAVA_HOME (if JAVA_HOME is not defined
in $DEFAULT)
31 JDK_DIRS="/usr/lib/jvm/java-7-oracle /usr/lib/jvm/java-6-openjdk
/usr/lib/jvm/java-6-sun /usr/lib/jvm/java-1.5.0-sun /usr/lib/j2sdk1.5-sun
/usr/lib/j2sdk1.5-ibm /usr/lib/jvm/java-1.7.0-openjdk-amd64 /usr/lib/jvm/java-1.7.0-
openjdk"

32 # Look for the right JVM to use
33 for jdir in $JDK_DIRS; do
34     if [ -r "$jdir/bin/java" -a -z "${JAVA_HOME}" ]; then
35         JAVA_HOME="$jdir"
36     fi
37 done
38 export JAVA_HOME
...

```

2. Modificare il file `/etc/init.d/raptorwebd` aggiungendo i valori in **grassetto**:

```

1 #!/bin/bash

...

25 ### END INIT INFO
26
27 ##
28 ## Added the JAVA_HOME detection, the Debian way --csr2 2012-07-19
29 ##

30 # The first existing directory is used for JAVA_HOME (if JAVA_HOME is not defined
in $DEFAULT)
31 JDK_DIRS="/usr/lib/jvm/java-7-oracle /usr/lib/jvm/java-6-openjdk
/usr/lib/jvm/java-6-sun /usr/lib/jvm/java-1.5.0-sun /usr/lib/j2sdk1.5-sun
/usr/lib/j2sdk1.5-ibm /usr/lib/jvm/java-1.7.0-openjdk-amd64 /usr/lib/jvm/java-1.7.0-
openjdk"

32 # Look for the right JVM to use
33 for jdir in $JDK_DIRS; do
34     if [ -r "$jdir/bin/java" -a -z "${JAVA_HOME}" ]; then
35         JAVA_HOME="$jdir"
36     fi
37 done
38 export JAVA_HOME

...

```

3. Modificare i file:

- `/opt/raptor/mua/conf/metadata.xml`: inserendo nome dell'organizzazione, contatto email del responsabile del MUA, un nome leggibile per il servizio (e.g.: UNIMO Raptor MUA Server) e l'entityId univoco del server così composto:
"https://raptor.example.com/raptor-mua"
- `/opt/raptor/web/conf/metadata.xml`: inserendo nome dell'organizzazione, contatto email del responsabile del Web, un nome leggibile per il servizio (e.g.: UNIMO Raptor Web Server) e l'entityId univoco del server così composto:
"https://raptor.example.com/raptor-web"
- `/opt/raptor/mua/conf/server.properties`: sostituendo i **"changeit"** con una password più sicura.

(Ogni altro cambiamento non è richiesto se lo Shibboleth IdP installato sul Raptor ICA usa i percorsi predefiniti della sua installazione)

4. Generare la chiave pubblica del Raptor MUA:

- `cd /opt/raptor/mua/keys`
- `keytool -genkeypair -alias raptormua -keystore /opt/raptor/mua/keys/raptor-mua.jks -storepass ##PASSWORD-M## -keypass ##PASSWORD-M## -dname "CN=`hostname -f`,ou=Raptor MUA Server,o=##YOUR-ORG##" -validity 7300 -keyalg RSA -keysize 2048`

(usare solo una password per entrambi i valori richiesti dal comando)

- `keytool -export -alias raptormua -keystore /opt/raptor/mua/keys/raptor-mua.jks -storepass ##PASSWORD-M## -file /opt/raptor/mua/keys/raptor-mua-public.crt`

5. Generare la chiave pubblica del Raptor WEB:

- `cd /opt/raptor/web/keys`
- `keytool -genkeypair -alias raptorweb -keystore /opt/raptor/web/keys/raptor-web.jks -storepass ##PASSWORD-W## -keypass ##PASSWORD-W## -dname "CN=`hostname -f`,ou=Raptor WEB Server,o=##YOUR-ORG##" -validity 7300 -keyalg RSA -keysize 2048`

(usare solo una password per entrambi i valori richiesti dal comando)

- `keytool -export -alias raptorweb -keystore /opt/raptor/web/keys/raptor-web.jks -storepass ##PASSWORD-W## -file /opt/raptor/web/keys/raptor-web-public.crt`

6. Importare il certificato del Raptor MUA nel Raptor WEB e viceversa:

- `keytool -import -keystore /opt/raptor/mua/keys/authorised-keys.jks -storepass ##PASSWORD-M## -alias raptorweb -file /opt/raptor/web/keys/raptor-web-public.crt`
- `keytool -import -keystore /opt/raptor/web/keys/authorised-keys.jks -storepass ##PASSWORD-W## -alias raptormua -file /opt/raptor/mua/keys/raptor-mua-public.crt`

7. Verificare il successo dell'importazione:

- `keytool -list -v -keystore /opt/raptor/mua/keys/authorised-keys.jks (##PASSWORD-M##)`
- `keytool -list -v -keystore /opt/raptor/web/keys/authorised-keys.jks (##PASSWORD-W##)`

8. Esportare il certificato del Raptor MUA sui diversi Raptor ICA:

- `scp /opt/raptor/mua/keys/raptor-mua-public.crt root@idp.example.it:/opt/raptor/ica/keys`

9. Cambiare la password dell'utente “**admin**” e aggiungere nuovi utenti aventi accesso all'interfaccia web:

1. Da terminale eseguire “echo -n '##nuova_password##' | md5sum -”
2. Aggiungere un nuovo utente:
 - vim /opt/raptor/web/conf/users.xml

```
<s:authentication-manager>
  <s:authentication-provider>
    <s:password-encoder hash="md5" />
    <s:user-service>
      <s:user name="admin" password="##NEW_ECHO_MD5_STRING##"
        authorities="ROLE_ADMIN" />
      <s:user name="user1" password="##NEW_ECHO_MD5_STRING_4_USER1##"
        authorities="ROLE_ADMIN" />
    </s:user-service>
  </s:authentication-provider>
</s:authentication-manager>
```

10. Modificare i valori “**changeit**” all'interno del file “/opt/raptor/mua/conf/event-release.xml” con **##PASSWORD-M##** e sostituire
 “<value>https://raptor-aggr.ja.net:8111/MUA/MultiUnitAggregator</value>” con
 “<value>https://##RAPTOR-MUA-FQDN##:8111/MUA/MultiUnitAggregator</value>”.

11. Configurare il Raptor MUA in modo che recuperi i metadati corretti:

- vim /opt/raptor/mua/conf/statistical-processors.xml

```
<bean
  class="org.opensaml.saml2.metadata.provider.FileBackedHTTPMetadataProvider">
  <constructor-arg index="0"
    value="http://www.garr.it/idem-metadata/edugain2idem-metadata-
sha256.xml" />
  <constructor-arg index="1" value="5000" />
  <constructor-arg index="2" value="conf/edugain2idem-metadata.xml" />
</bean>
```

12. Configurare il Raptor WEB in modo che comunichi correttamente col Raptor MUA cambiando ogni valore “**changeit**” in “**##PASSWORD-W##**” nel file “/opt/raptor/web/conf/mua-endpoints.xml”

13. Configurare Apache2 per eseguire l'applicazione Raptor su HTTPS:

- vim /etc/apache2/sites-available

```
ProxyPass / http://localhost:8112/
ProxyPassReverse / http://localhost:8112/
```

- cd /etc/apache2/mods-available ; a2enmod proxy_http
- cd /etc/apache2/sites-available ; a2ensite raptor
- service apache2 restart

14. Se presente un firewall locale alla macchina, configurarlo in modo che la porta 443 sia aperta:

- sudo iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT

15. Istanziare un nuovo database MySQL per il Raptor MUA:

- `sudo apt-get install mysql-server` (scegliere la **##PASSWORD-DB-MUA##**)
- `mysql -u root -p` (digitare **##PASSWORD-DB-MUA##**)
- `create database mua;`
- `grant all on mua.* to raptor@'localhost' identified by '##PASSWORD-RAPTOR##'`
- `flush privileges`
- `exit`

16. Configurare il MUA in modo che comunichi con il DB creato:

- `vim /opt/raptor/mua/conf/database.xml`

```

<!-- HIBERNATE DAO -->
<bean id="muaSessionFactory"
  class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
  <property name="dataSource">
    <!-- Choose the type of database you are using by altering the name
below to match the config at the bottom of this file, i.e. one of:
      * muac3p0DataSourceHSQLDB
      * muac3p0DataSourceMySQL
      * muac3p0DataSourceOracle
      * muac3p0DataSourcePostgres
    -->
    <!--ref bean="muac3p0DataSourceHSQLDB" /-->
    <ref bean="muac3p0DataSourceMySQL" />
  </property>
  <property name="hibernateProperties">
    <props>
      <!-- All of these hibernate.dialect properties should be com-
mented out apart from the one you want to use -->
      <!-- <prop key="hibernate.dialect"> org.hibernate.dialect.HSQL-
Dialect</prop>-->
      <prop key="hibernate.dialect"> org.hibernate.dialect.MySQL-
Dialect</prop>
      <!-- <prop key="hibernate.dialect"> org.hibernate.dialect.Ora-
cleDialect</prop> -->
      <!-- <prop key="hibernate.dialect"> org.hibernate.dialect.Post-
greSQLDialect</prop>-->
      <prop key="hibernate.show_sql">false</prop>
      <prop key="hibernate.hbm2ddl.auto">create</prop>
    </props>
  </property>
...
<!-- Users of MySQL can use the following c3p0 configuration as a template -->
<bean id="muac3p0DataSourceMySQL" class="com.mchange.v2.c3p0.ComboPooledData-
Source"
  destroy-method="close">
  <property name="driverClass">
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property name="jdbcUrl">
    <value>jdbc:mysql://localhost/mua</value>
  </property>
  <property name="user">
    <value>raptor</value>
  </property>
  <property name="password">
    <value>##PASSWORD-RAPTOR##</value>
  </property>

```

```

    <property name="properties">
      <props>
        <prop key="c3p0.acquire_increment">5</prop>
        <prop key="c3p0.idle_test_period">100</prop>
        <prop key="c3p0.max_size">100</prop>
        <prop key="c3p0.max_statements">0</prop>
        <prop key="c3p0.min_size">10</prop>
        <prop key="user">raptor</prop>
        <prop key="password">##PASSWORD-RAPTOR##</prop>
      </props>
    </property>
  </bean>

```

- `service raptordmuad start` (attendere 5/6 minuti)

17. Ripristinare il valore “**update**” nel file di configurazione del MUA:

- `vim /opt/raptor/mua/conf/database.xml`

```

<prop key="hibernate.hbm2ddl.auto">create</prop>
to
<prop key="hibernate.hbm2ddl.auto">update</prop>

```

- `service raptormuad restart`
- Aspettare 2 minuti
- `service raptorwebd start`
- Aspettare 3 minuti

18. Provate ad accedere al vostro Raptor Server dalla url: <https://raptor.example.org>

4.3 Scambio di Chiavi tra MUA e ICA

4.3.1 Raptor ICA - Importazione Chiave del MUA

- `cd /tmp ; chown root:root /tmp/raptor-mua-public.crt`
- `cd /opt/raptor/ica/keys`
- `keytool -import -keystore authorised-keys.jks -storepass ##PASSWORD-I## -alias raptormua -file /tmp/raptor-mua-public.crt`
- `rm /tmp/raptor-mua-public.crt`
- `service raptoricad restart`

4.3.2 Raptor MUA - Importazione Chiave del ICA

- `cd /tmp ; chown root:root /tmp/raptor-ica-public.crt`
- `cd /opt/raptor/mua/keys`
- `keytool -import -keystore authorised-keys.jks -storepass ##PASSWORD-M## -alias raptormua -file /tmp/raptor-ica-public.crt`
- `rm /tmp/raptor-ica-public.crt`
- `service raptormuad restart`
- aspetta 3 minuti
- `service raptorwebd restart`

5 Appendice A

Raptor è in grado di gestire anche informazioni aggiuntive degli utenti (e.g.: dipartimento, affiliazione) prelevandole dal proprio IdM. In questo tutorial mostreremo come fornire tali informazioni prelevandole da un DB SQL. Seguite le istruzioni:

- Creare un DB per la memorizzazione delle informazioni aggiuntive degli utenti

```
SET NAMES 'utf8';

SET CHARACTER SET utf8;

CREATE DATABASE IF NOT EXISTS identities CHARACTER SET=utf8;

GRANT ALL PRIVILEGES ON identities.* TO raptor@'localhost' identified by '##PASS-WORD-RAPTOR##';
FLUSH PRIVILEGES;

use identities;

CREATE TABLE IF NOT EXISTS Users
(
    id MEDIUMINT NOT NULL AUTO_INCREMENT,
    userID VARCHAR(255) NOT NULL,
    department VARCHAR(255) NOT NULL,
    eduPersonAffiliation VARCHAR(255) NOT NULL,
    PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET="utf8";
```

- `vim /opt/raptor/mua/conf/attribute-association.xml`

```
<bean id="attributeAssociationEngine" class="uk.ac.cardiff.raptor.event.expansion.AttributeAssociationEngine">
    <property name="attributeAssociationDefinitions">
        <list>
            <ref bean="shibPrincipalAttributeAssociationDefinition"/>
            <!--<ref bean="ezproxyPrincipalAttributeAssociationDefinition"/>-->
            <ref bean="shibResourceCategoryAttributeAssociationDefinition"/>
        </list>
    </property>
</bean>

....
<bean id="shibPrincipalAttributeAssociationDefinition" class="uk.ac.cardiff.raptor.event.expansion.PrincipalAttributeAssociationDefinition">
    <property
name="definationName"><value>ShibAssociationDefinition</value></property>
    <property name="subjectPrincipalField"><value>principalName</value></prop-
erty>
    <property name="dataConnector"><ref bean="databaseConnector"/></property>
    <!-- add the filter here, where [principal] is replaced by the actual prin-
cipal value extracted per event -->
    <property name="searchTemplate"><value>select * from Users where userID =
&apos;[principal]&apos;</value></property>
    <property name="enabled"><value>true</value></property>
    <property name="classToAdd"><value type="java.lang.Class">uk.ac.cardiff.mod-
```

```

el.event.auxiliary.PrincipalInformation</value></property>
  <property name="associateWithClass"><value>uk.ac.cardiff.model.event.Shibbo-
lethIdpAuthenticationEvent</value></property>
  <property name="lookupAttributes">
    <list>
      <bean class="uk.ac.cardiff.raptor.event.expansion.AttributeLookup">
        <property name="sourceAttributeName"><value>eduPersonAffilia-
tion</value></property>
        <property name="internalAttributeName"><value>affilia-
tion</value></property>
      </bean>
      <bean class="uk.ac.cardiff.raptor.event.expansion.AttributeLookup">
        <property name="sourceAttributeName"><value>depart-
ment</value></property>
        <property
name="internalAttributeName"><value>school</value></property>
      </bean>
    </list>
  </property>
</bean>
....
<!-- Example RDBMS Data connectors for postgres. But mysql is also possible.
-->
  <bean id="databaseConnector" class="uk.ac.cardiff.raptor.event.expansion.connec-
tor.RDBMSDataConnector">
    <property name="cacheResults"><value>true</value></property>
    <property name="cacheTimeoutMs"><value>86400000</value></property>
    <property name="dataSource">
      <bean id="dataSourceConnectionProperties" class="org.apache.commons.d-
bcp.BasicDataSource">
        <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
        <property name="url" value="jdbc:mysql://localhost/identities"/>
        <property name="username" value="raptor"/>
        <property name="password" value="##PASSWORD-RAPTOR##"/>
      </bean>
    </property>
  </bean>

```

- La proprietà “*searchTemplate*” deve contenere il filtro di ricerca/query SQL che restituisce gli utenti dal nuovo DB creato. Nella stringa di ricerca verrà sostituito a “[**principal**]” il valore del *principalName* contenuto nell'*idp-audit.log* inviato dal Raptor ICA al Raptor MUA.
- Una volta estratto il record-utente dal DB attraverso la query della proprietà “*searchTemplate*”, i valori delle colonne “**eduPersonAffiliation**” e “**department**” verranno assegnati alle variabili interne “*affiliation*” e “*school*” in modo da poter essere utilizzati per le statistiche.
- Aggiungete i seguenti beans per estendere le statistiche con l'affiliazione e il dipartimento degli utenti:

```
vim /opt/raptor/mua/conf/statistical-unit-custom.xml
```

```

  <bean id="authsPerSchool" class="uk.ac.cardiff.raptormua.engine.statistics.-
functions.GroupByFrequency">
    <property name="StatisticParameters">
      <bean class="uk.ac.cardiff.model.wsmodel.StatisticParameters">
        <property name="statisticType"><value>User</value></property>
        <property name="eventType"><value>uk.ac.cardiff.model.even-

```

```

t.ShibbolethIdpAuthenticationEvent</value></property>
    <property name="unitName"><value>Number of authentications per
school</value></property>
    <property name="startTime"><value>20140101T000000</value></prop-
erty>  <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss →
    <property name="endTime"><value>20150101T000000</value></prop-
erty>  <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss e.g. 20101020T225100Z →
    <property name="methodParams">
        <list>
            <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-
ter">
                <property name="value"><value>principalInforma-
tion.school</value></property>
            </bean>
        </list>
    </property>

    <property name="series">
        <list>
            <bean class="uk.ac.cardiff.model.report.Series">
                <property name="seriesLabel" value="Number of au-
thentications (ALL)"/>
            </bean>
            <bean class="uk.ac.cardiff.model.report.Series">
                <property name="seriesLabel" value="Number of au-
thentications (Undergraduate)"/>
            </bean>
            <property name="comparisonPredicate">
                <bean class="uk.ac.cardiff.model.sql.Compar-
isonPredicate">
                    <property name="compOp"
value="EQUAL"/>
                    <property name="fieldName"
value="principalInformation.affiliation"/>
                    <property name="value" value="U"/>
                </bean>
            </property>
        </list>
    </property>

    <property name="presentation">
        <bean class="uk.ac.cardiff.model.report.Presentation">
            <property name="graphTitle" value="Number of authentica-
tions per school"/>
            <property name="xAxisLabel" value="Time"/>
            <property name="yAxisLabel" value="Number"/>
        </bean>
    </property>
</bean>
</property>
<property name="attachProcessors">
    <list>
        <bean class="uk.ac.cardiff.model.wsmodel.ProcessorInformation">
            <property name="processorClass">
                <value>uk.ac.cardiff.raptormua.engine.statistics.processor.-
SortGroupsAlphabeticallyPostProcessor</value>
            </property>
            <property name="methodParameters">
                <list>
                    <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-

```

```

ter">
        <property name="parameterType" value="VALUE" />
        <property name="parameterName"
value="Ascending"></property>
        <property name="value">
            <value type="java.lang.Boolean">true</value>
        </property>
    </bean>
</list>
</property>
</bean>
</list>
</property>
</bean>

    <bean id="authsPerPersonAffiliation" class="uk.ac.cardiff.raptormua.engine.sta-
tistics.functions.GroupByFrequency">
        <property name="StatisticParameters">
            <bean class="uk.ac.cardiff.model.wsmodel.StatisticParameters">
                <property name="statisticType"><value>User</value></property>
                <property name="eventType"><value>uk.ac.cardiff.model.even-
t.ShibbolethIdpAuthenticationEvent</value></property>
                <property name="unitName"><value>Number of authentications per
affiliation type</value></property>
                <property name="startTime"><value>20140101T000000</value></prop-
erty> <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss →
                <property name="endTime"><value>20150101T000000</value></prop-
erty> <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss e.g. 20101020T225100Z →
                <property name="methodParams">
                    <list>
                        <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-
ter">
                            <property name="value"><value>principalInforma-
tion.affiliation</value></property>
                        </bean>
                    </list>
                </property>

                <property name="series">
                    <list>
                        <bean class="uk.ac.cardiff.model.report.Series">
                            <property name="seriesLabel" value="Number of au-
thentications"/>
                        </bean>
                    </list>
                </property>

                <property name="presentation">
                    <bean class="uk.ac.cardiff.model.report.Presentation">
                        <property name="graphTitle" value="Number of authentica-
tions per affiliation type"/>
                        <property name="xAxisLabel" value="Time"/>
                        <property name="yAxisLabel" value="Number"/>
                    </bean>
                </property>
            </bean>
        </property>
        <property name="attachProcessors">
            <list>
                <bean class="uk.ac.cardiff.model.wsmodel.ProcessorInformation">

```



```

        <property name="processorClass">
            <value>uk.ac.cardiff.raptormua.engine.statistics.processor.-
SortGroupsAlphabeticallyPostProcessor</value>
        </property>
        <property name="methodParameters">
            <list>
                <bean class="uk.ac.cardiff.model.wsmodel.MethodParame-
ter">
                    <property name="parameterType" value="VALUE" />
                    <property name="parameterName"
value="Ascending"></property>
                    <property name="value">
                        <value type="java.lang.Boolean">true</value>
                    </property>
                </bean>
            </list>
        </property>
    </bean>
</list>
</property>
</bean>

```

- `service raptormuad restart`
- Aspettare 3 minuti
- `service raptorwebd restart`
- Aspettare 5 minuti
- Da questo momento in poi gli utenti presenti nel DB, riconosciuti attraverso il **principal name** recuperato dagli *idp-audit.log*, saranno muniti di affiliazione e dipartimento permettendo un più elaborato livello di statistiche.



HOWTO Install Raptor MUA+WEB on Debian Linux

14 Gennaio 2015

Author: Marco Malavolti

Index

1) Introduction.....	20
2) Packages required.....	20
3) Configure Apache2 SSL.....	21
4) Installation.....	23
4.1) Download and Install the Raptor MUA and the Raptor WEB.....	23
4.2) Configure Raptor MUA e WEB.....	24
4.3) Key exchange between MUA and ICA.....	29
4.3.1) Raptor ICA – Raptor MUA key import.....	29
4.3.2) Raptor MUA – Raptor ICA key import.....	29
5) Appendix A.....	30

1 Introduction

This tutorial wants help the users to install a Raptor Server (MUA+WEB) on a Debian Linux (Wheezy). A Raptor Server collects and presents the informations provided by the Raptor Client (ICA) connected to it. We expect to find, on the Raptor ICA, a Shibboleth IdP.

2 Packages required

- ntp
- vim
- java (openjdk o sun-java)
- alien
- wget
- mysql-server

3 Configure Apache2 SSL

1. Became ROOT on the target machine:
 - `sudo su -`
2. Create the “/root/certificates” directory and insert into it the HTTPS certificate, the HTTP key and the Terena-chain provided by GARR:
 - `mkdir /root/certificates`
 - `mv /tmp/cert-server.pem /tmp/key-server.pem /tmp/Terena-Chain.pem /root/certificates`
3. Modify the configuration of Apache2:
 - `vim /etc/apache2/sites-available/default-ssl`

```
<VirtualHost _default_:443>
    ServerName raptor.example.org:443
    ServerAdmin admin@example.org
    ErrorLog ${APACHE_LOG_DIR}/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    SSLEngine On
    SSLProtocol all -SSLv2 -SSLv3 -TLSv1
    SSLCipherSuite "KEDH+AESGCM:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES256-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA256:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA256:AES256-GCM-SHA384:!3DES:!DES:!DHE-RSA-AES128-GCM-SHA256:!DHE-RSA-AES256-SHA:!EDE3:!EDH-DSS-CBC-SHA:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC-SHA:!EDH-RSA-DES-CBC3-SHA:!EXP-EDH-DSS-DES-CBC-SHA:!EXP-EDH-RSA-DES-CBC-SHA:!EXPORT:!MD5:!PSK:!RC4-SHA:!aNULL:!eNULL"
    SSLHonorCipherOrder on

    # Disable SSL Compression
    SSLCompression Off

    # Enable HTTP Strict Transport Security with a 2 year duration
    Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"

    SSLCertificateFile /root/certificates/cert-server.pem
    SSLCertificateKeyFile /root/certificates/key-server.pem
    SSLCertificateChainFile /root/certificates/Terena-Chain.pem

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>

    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>

    BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

    # MSIE 7 and newer should be able to use keepalive
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
```

```
</VirtualHost>
```

- `vim /etc/apache2/ports.conf`

```
#NameVirtualHost *:80  
#Listen 80
```

4. Enable SSL on the Apache2 server and restart the service:
 - `cd /etc/apache2/sites-available`
 - `a2enmod header ; a2enmod ssl ; a2ensite default-ssl`
 - `service apache2 restart`
5. Verify to get at least an “A” value on this site:
 - <https://www.ssllabs.com/ssltest/>

4 Installation

4.1 Download and Install the Raptor MUA and the Raptor WEB

1. Became ROOT on the target machine:
 - `sudo su -`
2. Install the packages required:
 - `apt-get install ntp vim openjdk-7-jre alien wget`
3. Download the packages of Raptor WEB and Raptor MUA into `/usr/local/src`:
 - `wget -c -nd`
`http://download.opensuse.org/repositories/home:/rhyssmith:/raptor/SLE_11/noarch/raptor-web-1.2.1-1.1.noarch.rpm -O /usr/local/src`
 - `wget -c -nd`
`http://download.opensuse.org/repositories/home:/rhyssmith:/raptor/SLE_11/noarch/raptor-mua-1.2.1-1.1.noarch.rpm -O /usr/local/src`
4. Convert the RPM packages downloaded into DEB packages with **alien**:
 - `cd /usr/local/src`
 - `alien --scripts raptor-web-1.2.1-1.1.noarch.rpm`
 - `alien --scripts raptor-mua-1.2.1-1.1.noarch.rpm`
5. Install the DEB packages created:
 - `dpkg -i raptor-web_1.2.1-2.1_all.deb`
 - `dpkg -i raptor-mua_1.2.1-2.1_all.deb`

4.2 Configure Raptor MUA e WEB

1. Modify `/etc/init.d/raptormuad` file by adding the bold values:

```
1 #!/bin/bash
...

25 ### END INIT INFO
26
27 ##
28 ## Added the JAVA_HOME detection, the Debian way --csr2 2012-07-19
29 ##
30 # The first existing directory is used for JAVA_HOME (if JAVA_HOME is not defined
in $DEFAULT)
31 JDK_DIRS="/usr/lib/jvm/java-7-oracle /usr/lib/jvm/java-6-openjdk
/usr/lib/jvm/java-6-sun /usr/lib/jvm/java-1.5.0-sun /usr/lib/j2sdk1.5-sun
/usr/lib/j2sdk1.5-ibm /usr/lib/jvm/java-1.7.0-openjdk-amd64 /usr/lib/jvm/java-1.7.0-
openjdk"

32 # Look for the right JVM to use
33 for jdir in $JDK_DIRS; do
34     if [ -r "$jdir/bin/java" -a -z "${JAVA_HOME}" ]; then
35         JAVA_HOME="$jdir"
36     fi
37 done
38 export JAVA_HOME

...
```


2. Modify the /etc/init.d/raptorwebd file by adding the bold values:

```

1 #!/bin/bash

...

25 ### END INIT INFO
26
27 ##
28 ## Added the JAVA_HOME detection, the Debian way --csr2 2012-07-19
29 ##

30 # The first existing directory is used for JAVA_HOME (if JAVA_HOME is not defined
in $DEFAULT)
31 JDK_DIRS="/usr/lib/jvm/java-7-oracle /usr/lib/jvm/java-6-openjdk
/usr/lib/jvm/java-6-sun /usr/lib/jvm/java-1.5.0-sun /usr/lib/j2sdk1.5-sun
/usr/lib/j2sdk1.5-ibm /usr/lib/jvm/java-1.7.0-openjdk-amd64 /usr/lib/jvm/java-1.7.0-
openjdk"

32 # Look for the right JVM to use
33 for jdir in $JDK_DIRS; do
34     if [ -r "$jdir/bin/java" -a -z "${JAVA_HOME}" ]; then
35         JAVA_HOME="$jdir"
36     fi
37 done
38 export JAVA_HOME

...

```

3. Modify the files:

- /opt/raptor/mua/conf/metadata.xml: insert the name of the Organization, the e-mail of the MUA's responsible, a readable name for the MUA Server (e.g.: UNIMO Raptor MUA Server) and the entityId unique of the MUA server like the follow:
"https://raptor.example.com/raptor-mua"
- /opt/raptor/web/conf/metadata.xml: insert the name of the Organization, the e-mail of the MUA's responsible, a readable name for the WEB Server (e.g.: UNIMO Raptor WEB Server) and the entityId unique of the WEB server like the follow:
"https://raptor.example.com/raptor-web"
- /opt/raptor/mua/conf/server.properties: replace the **"changeit"** with a different password.

(Any other change is not required if the Shibboleth IdP installed on the target machine uses the default paths)

4. Generate the public key for the Raptor MUA server:

- `cd /opt/raptor/mua/keys`
- `keytool -genkeypair -alias raptormua -keystore /opt/raptor/mua/keys/raptor-mua.jks -storepass ##PASSWORD-M## -keypass ##PASSWORD-M## -dname "CN=`hostname -f`,ou=Raptor MUA Server,o=##YOUR-ORG##" -validity 7300 -keyalg RSA -keysize 2048`

(use only one password for both values required for the command for simplicity)

- `keytool -export -alias raptormua -keystore /opt/raptor/mua/keys/raptor-mua.jks -storepass ##PASSWORD-M## -file /opt/raptor/mua/keys/raptor-mua-public.crt`

5. Generate the public key for the Raptor WEB server:

- `cd /opt/raptor/web/keys`
- `keytool -genkeypair -alias raptorweb -keystore /opt/raptor/web/keys/raptor-web.jks -storepass ##PASSWORD-W## -keypass ##PASSWORD-W## -dname "CN=`hostname -f`,ou=Raptor WEB Server,o=##YOUR-ORG##" -validity 7300 -keyalg RSA -keysize 2048`

(use only one password for both values required for the command for simplicity)

- `keytool -export -alias raptorweb -keystore /opt/raptor/web/keys/raptor-web.jks -storepass ##PASSWORD-W## -file /opt/raptor/web/keys/raptor-web-public.crt`

6. Import the Raptor MUA certificat into the Raptor WEB keystore and viceversa:

- `keytool -import -keystore /opt/raptor/mua/keys/authorised-keys.jks -storepass ##PASSWORD-M## -alias raptorweb -file /opt/raptor/web/keys/raptor-web-public.crt`
- `keytool -import -keystore /opt/raptor/web/keys/authorised-keys.jks -storepass ##PASSWORD-W## -alias raptormua -file /opt/raptor/mua/keys/raptor-mua-public.crt`

7. Verify the success of the import:

- `keytool -list -v -keystore /opt/raptor/mua/keys/authorised-keys.jks (##PASSWORD-M##)`
- `keytool -list -v -keystore /opt/raptor/web/keys/authorised-keys.jks (##PASSWORD-W##)`

8. Export the Raptor MUA's certificate and send it to the Raptor ICA server:

- `scp /opt/raptor/mua/keys/raptor-mua-public.crt root@idp.example.it:/opt/raptor/ica/keys`

9. Change the password of the user “**admin**” and, if you want, add other users able to access to the web interface by following this steps:

1. Execute “`echo -n '##nuova_password##' | md5sum -`”
2. Add the new user “**user1**”:
 - `vim /opt/raptor/web/conf/users.xml`

```
<s:authentication-manager>
  <s:authentication-provider>
    <s:password-encoder hash="md5" />
    <s:user-service>
      <s:user name="admin" password="##NEW_ECHO_MD5_STRING##"
        authorities="ROLE_ADMIN" />
      <s:user name="user1" password="##NEW_ECHO_MD5_STRING_4_USER1##"
        authorities="ROLE_ADMIN" />
    </s:user-service>
  </s:authentication-provider>
</s:authentication-manager>
```

10. Modify the values “**changeit**” into the file “`/opt/raptor/mua/conf/event-release.xml`” with **##PASSWORD-M##** and replace

“`<value>https://raptor-aggr.ja.net:8111/MUA/MultiUnitAggregator</value>`” with
“`<value>https://##RAPTOR-MUA-FQDN##:8111/MUA/MultiUnitAggregator</value>`”.

11. Configure the Raptor MUA to retrieve the IDEM Metadata:

- `vim /opt/raptor/mua/conf/statistical-processors.xml`

```
<bean
  class="org.opensaml.saml2.metadata.provider.FileBackedHTTPMetadataProvider">
  <constructor-arg index="0"
    value="http://www.garr.it/idem-metadata/edugain2idem-metadata-
sha256.xml" />
  <constructor-arg index="1" value="5000" />
  <constructor-arg index="2" value="conf/edugain2idem-metadata.xml" />
</bean>
```

12. Configure the Raptor WEB by changing each “**changeit**” value into “**##PASSWORD-W##**” on “`/opt/raptor/web/conf/mua-endpoints.xml`” file.

13. Configure ProxyPass on Apache2 to execute correctly the Raptor application on HTTPS:

- `vim /etc/apache2/sites-available`

```
ProxyPass / http://localhost:8112/
ProxyPassReverse / http://localhost:8112/
```

- `cd /etc/apache2/mods-available ; a2enmod proxy_http`
- `cd /etc/apache2/sites-available ; a2ensite raptor`
- `service apache2 restart`

14. If present, open the port 443 on your firewall:

- `sudo iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT`

15. Create a new MySQL database for the Raptor MUA:

- `sudo apt-get install mysql-server` (choice the **##PASSWORD-DB-MUA##**)
- `mysql -u root -p` (enter the **##PASSWORD-DB-MUA##**)
- `create database mua;`
- `grant all on mua.* to raptor@'localhost' identified by '##PASSWORD-RAPTOR##'`
- `flush privileges`
- `exit`

16. Connect Raptor MUA to the new DB created:

- `vim /opt/raptor/mua/conf/database.xml`

```

<!-- HIBERNATE DAO -->
<bean id="muaSessionFactory"
  class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
  <property name="dataSource">
    <!-- Choose the type of database you are using by altering the name
below to match the config at the bottom of this file, i.e. one of:
      * muac3p0DataSourceHSQLDB
      * muac3p0DataSourceMySQL
      * muac3p0DataSourceOracle
      * muac3p0DataSourcePostgres
    -->
    <!--ref bean="muac3p0DataSourceHSQLDB" /-->
    <ref bean="muac3p0DataSourceMySQL" />
  </property>
  <property name="hibernateProperties">
    <props>
      <!-- All of these hibernate.dialect properties should be com-
mented out apart from the one you want to use -->
      <!-- <prop key="hibernate.dialect"> org.hibernate.dialect.HSQL-
Dialect</prop>-->
      <prop key="hibernate.dialect"> org.hibernate.dialect.MySQL-
Dialect</prop>
      <!-- <prop key="hibernate.dialect"> org.hibernate.dialect.Ora-
cleDialect</prop> -->
      <!-- <prop key="hibernate.dialect"> org.hibernate.dialect.Post-
greSQLDialect</prop>-->
      <prop key="hibernate.show_sql">>false</prop>
      <prop key="hibernate.hbm2ddl.auto">create</prop>
    </props>
  </property>
...

<!-- Users of MySQL can use the following c3p0 configuration as a template -->
<bean id="muac3p0DataSourceMySQL" class="com.mchange.v2.c3p0.ComboPooledData-
Source"
  destroy-method="close">
  <property name="driverClass">
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property name="jdbcUrl">
    <value>jdbc:mysql://localhost/mua</value>
  </property>
  <property name="user">
    <value>raptor</value>
  </property>
  <property name="password">
    <value>##PASSWORD-RAPTOR##</value>
  </property>

```

```

    <property name="properties">
      <props>
        <prop key="c3p0.acquire_increment">5</prop>
        <prop key="c3p0.idle_test_period">100</prop>
        <prop key="c3p0.max_size">100</prop>
        <prop key="c3p0.max_statements">0</prop>
        <prop key="c3p0.min_size">10</prop>
        <prop key="user">raptor</prop>
        <prop key="password">##PASSWORD-RAPTOR##</prop>
      </props>
    </property>
  </bean>

```

- `service raptormuad start` (wait 5/6 minutes)

17. Recover the value “**update**” into the following configuration file of MUA Server:

- `vim /opt/raptor/mua/conf/database.xml`

```

<prop key="hibernate.hbm2ddl.auto">create</prop>
to
<prop key="hibernate.hbm2ddl.auto">update</prop>

```

- `service raptormuad restart`
- Wait 2 minutes
- `service raptorwebd start`
- Wait 3 minutes

18. Try to access the Raptor Server by opening: <https://raptor.example.org>

4.3 Key exchange between MUA and ICA

4.3.1 Raptor ICA – Raptor MUA key import

- `cd /tmp ; chown root:root /tmp/raptor-mua-public.crt`
- `cd /opt/raptor/ica/keys`
- `keytool -import -keystore authorised-keys.jks -storepass ##PASSWORD-I## -alias raptormua -file /tmp/raptor-mua-public.crt`
- `rm /tmp/raptor-mua-public.crt`
- `service raptoricad restart`

4.3.2 Raptor MUA – Raptor ICA key import

- `cd /tmp ; chown root:root /tmp/raptor-ica-public.crt`
- `cd /opt/raptor/mua/keys`
- `keytool -import -keystore authorised-keys.jks -storepass ##PASSWORD-M## -alias raptormua -file /tmp/raptor-ica-public.crt`
- `rm /tmp/raptor-ica-public.crt`
- `service raptormuad restart`
- wait 2 minutes
- `service raptorwebd restart`
- wait 3 minute before access the web application

5 Appendix A

Raptor is able to also manage other additional information of users (like department, affiliation) by retrieving from an IdM when requested. Into this tutorial we will create a new SQL DB to manage the additional department and affiliation of the users. Follow this steps:

- Create a new SQL DB able to store the additional information on the users:

```
SET NAMES 'utf8';

SET CHARACTER SET utf8;

CREATE DATABASE IF NOT EXISTS identities CHARACTER SET=utf8;

GRANT ALL PRIVILEGES ON identities.* TO raptor@'localhost' identified by '##PASS-WORD-RAPTOR##';
FLUSH PRIVILEGES;

use identities;

CREATE TABLE IF NOT EXISTS Users
(
  id MEDIUMINT NOT NULL AUTO_INCREMENT,
  userID VARCHAR(255) NOT NULL,
  department VARCHAR(255) NOT NULL,
  eduPersonAffiliation VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET="utf8";
```

- `vim /opt/raptor/mua/conf/attribute-association.xml`

```
<bean id="attributeAssociationEngine" class="uk.ac.cardiff.raptor.event.expansion.AttributeAssociationEngine">
  <property name="attributeAssociationDefinitions">
    <list>
      <ref bean="shibPrincipalAttributeAssociationDefinition"/>
      <!--<ref bean="ezproxyPrincipalAttributeAssociationDefinition"/>-->
      <ref bean="shibResourceCategoryAttributeAssociationDefinition"/>
    </list>
  </property>
</bean>

....
<bean id="shibPrincipalAttributeAssociationDefinition" class="uk.ac.cardiff.raptor.event.expansion.PrincipalAttributeAssociationDefinition">
  <property
name="definationName"><value>ShibAssociationDefinition</value></property>
  <property name="subjectPrincipalField"><value>principalName</value></property>
  <property name="dataConnector"><ref bean="databaseConnector"/></property>
  <!-- add the filter here, where [principal] is replaced by the actual principal value extracted per event -->
  <property name="searchTemplate"><value>select * from Users where userID =
&apos;[principal]&apos;;</value></property>
  <property name="enabled"><value>true</value></property>
  <property name="classToAdd"><value type="java.lang.Class">uk.ac.cardiff.mod-
```

```

el.event.auxiliary.PrincipalInformation</value></property>
  <property name="associateWithClass"><value>uk.ac.cardiff.model.event.ShibbolethIdpAuthenticationEvent</value></property>
  <property name="lookupAttributes">
    <list>
      <bean class="uk.ac.cardiff.raptor.event.expansion.AttributeLookup">
        <property name="sourceAttributeName"><value>eduPersonAffiliation</value></property>
        <property name="internalAttributeName"><value>affiliation</value></property>
      </bean>
      <bean class="uk.ac.cardiff.raptor.event.expansion.AttributeLookup">
        <property name="sourceAttributeName"><value>department</value></property>
        <property name="internalAttributeName"><value>school</value></property>
      </bean>
    </list>
  </property>
</bean>
....
<!-- Example RDBMS Data connectors for postgres. But mysql is also possible.
-->
  <bean id="databaseConnector" class="uk.ac.cardiff.raptor.event.expansion.connector.RDBMSDataConnector">
    <property name="cacheResults"><value>true</value></property>
    <property name="cacheTimeoutMs"><value>86400000</value></property>
    <property name="dataSource">
      <bean id="dataSourceConnectionProperties" class="org.apache.commons.dbcp.BasicDataSource">
        <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
        <property name="url" value="jdbc:mysql://localhost/identities"/>
        <property name="username" value="raptor"/>
        <property name="password" value="##PASSWORD-RAPTOR##"/>
      </bean>
    </property>
  </bean>

```

- The “*searchTemplate*” property needs to reflect the SQL query you will use to retrieve users from your Identity Management system. The value of “[**principal**]” will be replaced with the *principalName* retrieved from the *idp-audit.log* sent by Raptor ICA to Raptor MUA.
- Once the record of user is extracted, the values of the column “**eduPersonAffiliation**” and “**department**” will be assigned to the Raptor internal variable called “affiliation” and “school” to be used for statistics.
- Add the following beans to extend the statistics collected from Raptor:

```
vim /opt/raptor/mua/conf/statistical-unit-custom.xml
```

```

  <bean id="authsPerSchool" class="uk.ac.cardiff.raptormua.engine.statistics.functions.GroupByFrequency">
    <property name="StatisticParameters">
      <bean class="uk.ac.cardiff.model.wsmodel.StatisticParameters">
        <property name="statisticType"><value>User</value></property>
        <property name="eventType"><value>uk.ac.cardiff.model.event.ShibbolethIdpAuthenticationEvent</value></property>
      </bean>
    </property>
  </bean>

```

```

        <property name="unitName"><value>Number of authentications per
school</value></property>
        <property name="startTime"><value>20140101T000000</value></prop-
erty>  <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss →
        <property name="endTime"><value>20150101T000000</value></prop-
erty>  <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss e.g. 20101020T225100Z →
        <property name="methodParams">
            <list>
                <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-
ter">
                    <property name="value"><value>principalInforma-
tion.school</value></property>
                </bean>
            </list>
        </property>

        <property name="series">
            <list>
                <bean class="uk.ac.cardiff.model.report.Series">
                    <property name="seriesLabel" value="Number of au-
thentications (ALL)"/>
                </bean>
                <bean class="uk.ac.cardiff.model.report.Series">
                    <property name="seriesLabel" value="Number of au-
thentications (Undergraduate)"/>
                    <property name="comparisonPredicate">
                        <bean class="uk.ac.cardiff.model.sql.Compar-
isonPredicate">
                            <property name="compOp"
value="EQUAL"/>
                            <property name="fieldName"
value="principalInformation.affiliation"/>
                            <property name="value" value="U"/>
                        </bean>
                    </property>
                </bean>
            </list>
        </property>

        <property name="presentation">
            <bean class="uk.ac.cardiff.model.report.Presentation">
                <property name="graphTitle" value="Number of authentica-
tions per school"/>
                <property name="xAxisLabel" value="Time"/>
                <property name="yAxisLabel" value="Number"/>
            </bean>
        </property>
    </bean>
</property>
<property name="attachProcessors">
    <list>
        <bean class="uk.ac.cardiff.model.wsmodel.ProcessorInformation">
            <property name="processorClass">
                <value>uk.ac.cardiff.raptormua.engine.statistics.processor.-
SortGroupsAlphabeticallyPostProcessor</value>
            </property>
            <property name="methodParameters">
                <list>
                    <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-
ter">

```



```

        <property name="parameterType" value="VALUE" />
        <property name="parameterName"
value="Ascending"></property>
        <property name="value">
            <value type="java.lang.Boolean">true</value>
        </property>
    </bean>
</list>
</property>
</bean>
</list>
</property>
</bean>

    <bean id="authsPerPersonAffiliation" class="uk.ac.cardiff.raptormua.engine.sta-
tistics.functions.GroupByFrequency">
        <property name="StatisticParameters">
            <bean class="uk.ac.cardiff.model.wsmodel.StatisticParameters">
                <property name="statisticType"><value>User</value></property>
                <property name="eventType"><value>uk.ac.cardiff.model.even-
t.ShibbolethIdpAuthenticationEvent</value></property>
                <property name="unitName"><value>Number of authentications per
affiliation type</value></property>
                <property name="startTime"><value>20140101T000000</value></prop-
erty> <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss →
                <property name="endTime"><value>20150101T000000</value></prop-
erty> <!-- presently only ddMMyyy or yyyyMMdd'T'HHmmss e.g. 20101020T225100Z →
                <property name="methodParams">
                    <list>
                        <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-
ter">
                            <property name="value"><value>principalInforma-
tion.affiliation</value></property>
                        </bean>
                    </list>
                </property>

                <property name="series">
                    <list>
                        <bean class="uk.ac.cardiff.model.report.Series">
                            <property name="seriesLabel" value="Number of au-
thentications"/>
                        </bean>
                    </list>
                </property>

                <property name="presentation">
                    <bean class="uk.ac.cardiff.model.report.Presentation">
                        <property name="graphTitle" value="Number of authentica-
tions per affiliation type"/>
                        <property name="xAxisLabel" value="Time"/>
                        <property name="yAxisLabel" value="Number"/>
                    </bean>
                </property>
            </bean>
        </property>
        <property name="attachProcessors">
            <list>
                <bean class="uk.ac.cardiff.model.wsmodel.ProcessorInformation">
                    <property name="processorClass">

```

```

        <value>uk.ac.cardiff.raptormua.engine.statistics.processor.-
SortGroupsAlphabeticallyPostProcessor</value>
      </property>
      <property name="methodParameters">
        <list>
          <bean class="uk.ac.cardiff.model.wsmodel.MethodParamete-
ter">
            <property name="parameterType" value="VALUE" />
            <property name="parameterName"
value="Ascending"></property>
            <property name="value">
              <value type="java.lang.Boolean">true</value>
            </property>
          </bean>
        </list>
      </property>
    </bean>
  </list>
</property>
</bean>

```

- `service raptormuad restart`
- Wait 3 minutes
- `service raptorwebd restart`
- Wait 5 minutes
- Now each user recognized to be into the DB via his principal name will be enriched with the **affiliation** and **department** into the statistics.